Opportunistic Forwarding with Partial Centrality

Peiyan Yuan*, Huadong Ma*, Xiang-Yang Li†‡, Shaojie Tang†, Xufei Mao‡
*Department of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China
†Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
‡TNLIST, School of Software, Tsinghua University, Beijing, China

***Abstract*—In opportunistic networks, the use of social metrics (e.g., *degree*, *closeness* and *betweenness* centrality) of human mobility network, has recently been shown to be an effective solution to improve the performance of opportunistic forwarding algorithms. Most of the current social-based forwarding schemes exploit some globally defined node centrality, resulting in a bias towards the most popular nodes. However, these nodes may not be appropriate relay candidates for some target nodes, because they may have low importance relative to these subsets of target nodes. In this paper, to improve the opportunistic forwarding efficiency, we exploit the relative importance (called *partial centrality*) of a node with respect to a group of nodes. We design a new opportunistic forwarding scheme, *opportunistic forwarding with partial centrality* (OFPC), and theoretically quantify the influence of the partial centrality on the data forwarding performance using graph spectrum. By applying our scheme on three real opportunistic networking scenarios, our extensive evaluations show that our scheme achieves significantly better mean delivery delay and cost compared to the state-of-the-art works, while achieving delivery ratios sufficiently close to those by Epidemic under different TTL requirements.**

I. Introduction

The paradigm of opportunistic forwarding has been proposed to serve emerging wireless networking applications, where nodes experience intermittent connectivity [1] [2]. In such scenarios, to transmit messages to a distant destination under a given delay bound, node mobility is exploited to let nodes broker information exchange between disjoined parts of the network [3]. Therefore, the main challenge for opportunistic forwarding is to make an effective forwarding decision, such that the chosen relays have the best cumulative probability to the destination within the delay bound.

In most of the early works, due to the unstable end-to-end path and lack of global knowledge of the network topology, data forwarding decisions are generally made by adopting various heuristics, such as inferring the likelihood of forwarding the message (e.g., [4] [5] [6]), employing the contact locations (e.g., [7]), or focusing on the contact frequencies [8]. Obviously, these solutions guarantee packet delivery based on the prediction of *physical contact metrics*[1] of nodes. We argue that such solutions are not cost effective for opportunistic scenarios, since these simple metrics only reflect one facet of the underlying mobility process. On the other hand, with the recent popularization of personal hand-held mobile devices, human walks heavily affect the network performance [9] [10], e.g., devices may lose connection when people move around (in the rest of this paper, without loss of generality, we use the terms "people" and "node" interchangeably). We believe that the *social contact metrics* achieved by the complex network analysis [11] capture the inherent characteristics of the network structure, and are less volatile than the *physical contact metrics*. Motivated by the above observation, in this paper, we focus on integrating social metrics into the opportunistic forwarding algorithms. This design turns out to be critical while challenging, especially in an intermittently connected environment.

Recently, there are a few attempts to explicitly make use of the social metrics to formulate the opportunistic forwarding decision. Among them, SimBet [12], Bubble [13] and PeopleRank [14] are the three most recent works. Although the detailed forwarding schemes may differ, all of them are motivated by the following two important observations from society: 1) people with closer relationship tend to reside in *communities* and 2) people within a community may have different *popularity*. As such, the increasingly "popular" or "central" nodes are more probably chosen as carriers to relay messages between disconnected communities [14] [15], until a node belonging to the same community with the destination is reached [12] [13]. Intuitively, information about community structure and node popularity enables them to outperform well-known opportunistic forwarding algorithms that are not explicitly "social based".

Nevertheless, we notice that all of the three protocols prefer to use global measures of node centrality (e.g., exploiting ego networks in [12], betweenness centrality in [13] and PageRank [16] algorithm in [14]), in that each node is ranked with respect to all other nodes in the network. We argue that those popular nodes possibly with high global node centrality may not be the appropriate relay candidates, due to the fact that such nodes may have low importance relative to a specific subset of nodes, where the destination belongs. Interestingly, those nodes with low global node centrality but exhibiting high relative importance to the community partners of destination bear most weight on routing performance. In other words, this relative importance provides fine-grained relations among nodes. Thus, it helps to make informed forwarding decisions (e.g., a node is just the desired relay, if it exhibits a highly relative importance to the destination's community partners).

To this end, we first employ the partial centrality metric to measure the relative importance of a node with respect to such

[1]In this paper, we use the term "physical contact metrics" to denote contact pattern of a node with others such as contact time/frequency/location, and the term "social contact metrics" to denote those of similarity, centrality or community etc.

nodes within a community. We then develop an opportunistic forwarding scheme by jointly considering the partial centrality metric and the community structure, to improve the opportunistic forwarding efficiency. We summarize our contributions as follows:

- We evaluate the performance of opportunistic routing based on the partial centrality metric. To the best of our knowledge, this is the first attempt to integrate this social metric into opportunistic routing.
- We propose an online method to compute node's partial centrality in a distributed fashion, which makes our work more applicable. We also detect the overlapped community structure by effectively distinguishing the bridging nodes from other nodes, and exploit the community structure to label the community partners of destination.
- We formulate the strength of relationship between nodes as a Decayed Sum Problem [18], and use a Decayed Aggregation Graph (DAG) to model the dynamic of network topology.
- We implement OFPC and compare it to several state-of-the-art works through three real opportunistic networking scenarios. Our extensive evaluation results show that, overall, the OFPC outperforms other solutions, especially in terms of mean delivery delay and cost. For example, it achieves up to a 70% improvement in mean delivery delay over Prophet [4] and 40% over Bubble [13], and has a reduction of cost by up to 2 and 3 factors compared to Bubble and Prophet respectively in one network example.

We organize the remainder of this paper as follows. Section II overviews the problem and network model. Section III describes the forwarding scheme. In Section IV, we make a performance evaluation. After briefly reviewing the related work in Section V, we draw our conclusions in Section VI.

II. PRELIMINARIES

A. Centrality and Partial Centrality Metrics

Node centrality reflects the importance of a node relative to all other nodes in the network (i.e., how popular a person is within a social network), while node partial centrality measures the relative importance of a node with respect to such nodes within a group. Freeman [19] proposed three most widely used methods to estimate centrality, called degree, closeness and betweenness measures. We here take the degree measure as an example to illustrate the difference between centrality metric and partial centrality metric. Degree centrality is measured as the number of one-hop neighbors of a given node u, which reflects the direct relationship between the node u and its neighbors. In general, a node can directly contact with more other nodes, if it has a higher degree centrality. Node u's degree centrality is counted as:

$$D_C^u = \sum_{v=1, v\neq u}^{n} \delta_{uv} \tag{1}$$

where n is the number of nodes in the network and $\delta_{uv} = 1$ if node v is one of neighbors of node u, otherwise, $\delta_{uv} = 0$.

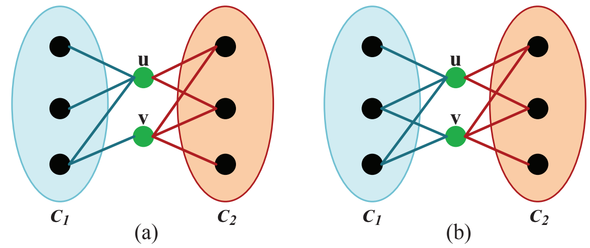


Fig. 1. Centrality and partial centrality metrics in two different situations: (a) $D_C^u = 5$ and $D_C^v = 4$; (b) $D_C^u = D_C^v = 5$. And in both situations, $D_{PC}^{(u,\mathcal{C}_2)} = 2$ and $D_{PC}^{(v,\mathcal{C}_2)} = 3$ (suppose the destination belongs to the community $\mathcal{C}_2$).

Similar to Eq. (1), a node's partial centrality is

$$D_{PC}^{(u,\mathcal{C}_k)} = \sum_{v=1, v\neq u}^{\|\mathcal{C}_k\|} \delta_{uv} \tag{2}$$

where $\| \mathcal{C}_k \|$ is the number of nodes in community $\mathcal{C}_k$ and $\delta_{uv} = 1$ if and only if 1) node v belongs to community $\mathcal{C}_k$ and 2) node v is one of neighbors of node u.

The main difference between the centrality metric and that of the partial centrality is illustrated in Fig.1, where node u needs to make a forwarding decision when it encounters node v. Node u will not forward the messages destined to community $\mathcal{C}_2$ to node v according to the traditional centrality-based forwarding schemes [12] [13] [14], this is mainly because that the centrality of node u is bigger than or equal to that of node v in the two situations, respectively. Whereas, if the forwarding schemes were guided by the partial centrality metric, node v would be a better relay since its partial centrality is bigger than that of node u.

B. Network Model

In this paper, we model an opportunistic network as a Decayed Aggregation Graph (DAG) $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. Let $W(t) = (w_{uv}(t))_{n\times n}$ denote its adjacency matrix and $N_{uv}(t)$=$\{(on_i, off_i),\ i$=1,2,...,$N\}$ denote the contact series between nodes u and v at moment t, where the tuple (on_i, off_i) denotes the start moment and end moment of the ith contact respectively, and N is the number of contacts. We formulate the computing strength of relationship between nodes (i.e., the value of $w_{uv}(t)$) as a Decayed Sum Problem.

Definition 1 (Decayed Sum): Given the contact series $N_{uv}(t)$, the goal is to estimate the decayed sum at any current time T

$$w_{uv}(T) = \sum_{i=1}^{N} f(i) g(T - off_i) \tag{3}$$

where $f(i) = off_i - on_i$ denotes the ith contact duration and $g(T - off_i)$ denotes the decayed function. In this paper, we set $g(T - off_i) = e^{(-\beta\times(T - off_i))}$, since the inter-contact time between nodes generally follows an exponential decay [20]. Hence, the Eq. (3) can be reformulated as

$$w_{uv}(T) = \sum_{i=1}^{N} (off_i - on_i) e^{-\beta\times(T - off_i)} \tag{4}$$

We next analyze the space complexity of DAG. Obviously, exact tracking of $w_{uv}(T)$ needs $\Theta(N)$ storage bits. Considering scalability issue (in general, $N \gg n$), we should further reduce the storage overhead while keeping the same calculation precision.

Let $h(t) = off_i - on_i$ if and only if t equals to off_i, otherwise, $h(t) = 0$, we obtain the following lemma.

Lemma 1: In a continuous interval $[0, T]$, the Eq. (4) is equivalent to the following Eq. (5)

$$w_{uv}(T) = \sum_{t \le T} h(t) e^{-\beta(T-t)} \tag{5}$$

Proof: Let us split the interval $[0, T]$ into two disjoined parts T_1 and T_2, where $T_1 = \bigcup_{i=1}^{N} t_i$ ($t_i = [on_i, off_i]$) and $T_1 \cup T_2 = [0, T]$. We have

$$\sum_{t \le T} h(t)e^{-\beta(T-t)} = \sum_{t \in T_1} h(t)e^{-\beta(T-t)} + \sum_{t \in T_2} h(t)e^{-\beta(T-t)}$$

For any $t \in T_2$, since $t \ne off_i$ (note that $off_i \in T_1$ and $T_1 \cap T_2 \equiv \emptyset$), we have $h(t) = 0$. Hence,

$$\begin{aligned} & \sum_{t \le T} h(t)e^{-\beta(T-t)} = \sum_{t \in T_1} h(t)e^{-\beta(T-t)} \\ = \quad & \sum_{i=1}^{N} \sum_{t \in t_i} h(t)e^{-\beta(T-t)} = \sum_{i=1}^{N} (off_i - on_i)e^{-\beta(T-off_i)} \\ = \quad & w_{uv}(T) \end{aligned}$$

∎

Theorem 1: At each time slot $t = 0, 1, 2, ..., T$, the value of $w_{uv}(T)$ can be maintained easily using

$$w_{uv}(T) = h(T) + e^{-\beta} w_{uv}(T-1) \tag{6}$$

Proof: From the *Lemma 1*, we have

$$\begin{aligned} w_{uv}(T) &= \sum_{t \le T} h(t)e^{-\beta(T-t)} \\ &= h(T)e^{-\beta(T-T)} + \sum_{t \le T-1} h(t)e^{-\beta(T-t)} \\ &= h(T) + \sum_{t \le T-1} h(t)e^{-\beta(T-1-t+1)} \\ &= h(T) + e^{-\beta} \sum_{t \le T-1} h(t)e^{-\beta(T-1-t)} \\ &= h(T) + e^{-\beta} w_{uv}(T-1) \end{aligned}$$

∎

From *Theorem 1*, each node only requires a single counter to exactly track the relationship between itself and any other node, which forms the row vector w_u of matrix W. As such, when two nodes meet each other, they can update their own matrix W by swapping such row vectors. Note that to keep the latest row vector w_u of matrix W, each node carries a list $Recent\_Time(n)$ as an indicator to record the last time when the corresponding $w_u (u = 1, 2, ..., n)$ was updated. Based on this indicator, they only need to swap the latest w_u to each other together with the $Recent\_Time(n)$. After that, they update their own W and $Recent\_Time(n)$, respectively.

III. RELAYING ALGORITHM

In this section, we first present our forwarding scheme OFPC in Section III.A, and then discuss how to quantify the partial centrality metric and detect the overlapped community structure in Section III.B and Section III.C, respectively.

A. Opportunistic Forwarding with Partial Centrality

We here present the OFPC algorithm. OFPC combines the knowledge of node partial centrality and that of overlapped community structure to make informed forwarding decisions. There are two intuitions behind this algorithm. First, the same person plays different social roles relative to different groups. Hence, one component of OFPC is to forward messages to nodes with higher partial centrality metrics to the destination communities than the current relay. Second, people show different social behaviors in society. Some tend to form one clique in their social lives. Others like to join multiple cliques. What is more, few people prefer to stay at home. Therefor, the other component of OFPC is to make different forwarding decisions based on the various types of nodes. The two components together form the algorithm. For this algorithm, we classify nodes into three categories 1) strong nodes (nodes only belonging to one community), 2) bridging nodes (nodes belonging to multiple communities) and 3) noise nodes (nodes not belonging to any community). Please refer to Section III.C for more formal definitions.

We next describe the baseline implementation of OFPC. Take node u and node v as samples. Suppose node u meets node v, for any message m that u carries, if its destination m_d is node v, node u delivers it to node v and removes it from u's message queue. Otherwise, if node v does not hold this message, node u makes different forwarding decisions based on the categories they belong to.

(1) Node v is a noise node: Node u does not forward the message m to node v.

(2) Node u is a noise node, but node v is a strong or bridging node: Node u forwards m to node v and deletes m from its buffer.

(3) Neither u nor v is a noise node: In this situation, if the message m has not been delivered to the community that the destination belongs to, it is forwarded to such nodes with higher partial centrality metrics (relative to the community partners of destination) than the current relay, until it reaches a node which shares a community with the destination node. Then the message is only forwarded to the community members with higher partial centrality metrics until the destination receives it or it expires. Furthermore, in order to further reduce cost, the original carriers can clear m from their buffer whenever m enters into the community[2]. Algorithm 1 outlines the above process, where ϕ denotes the $null\ set$, PC_u is the partial centrality of node u, $Com(u)$ denotes node u's set of

[2]When node v carries m as well, and node v is one of the community partners of the destination, node u can delete m from its buffer. Note that to prevent the situation where node u occasionally moves out of the same community, node u deletes m only $w_{um_d} < w_{vm_d}$ and node v carries m (as shown in *line 16, Algorithm 1*)

community labels and $(Com(u) \wedge Com(m_d) \neq \phi)$ denotes two nodes u and m_d belong to the same community, otherwise, they do not share one community.

Algorithm 1 OFPC, pseudo-code of node u

upon meeting up node v do
for any message m in u's queue **do**
  if $(m \notin v.queue)$ **then**
    if (neither u nor v is a noise node) **then**
      if $(Com(u) \wedge Com(m_d) == \phi)$ and $(Com(v) \wedge Com(m_d) == \phi)$ and $(PC_u < PC_v)$ **then**
        $m \rightarrow v$
      end if
      if $(Com(u) \wedge Com(m_d) == \phi)$ and $(Com(v) \wedge Com(m_d) \neq \phi)$ **then**
        $m \rightarrow v$
      end if
      if $(Com(u) \wedge Com(m_d) \neq \phi)$ and $(Com(v) \wedge Com(m_d) \neq \phi)$ and $(PC_u < PC_v)$ **then**
        $m \rightarrow v$
      end if
    end if
  end if
  if $(m \in v.queue)$ and $(Com(v) \wedge Com(m_d) \neq \phi)$ and $(Com(u) \wedge Com(m_d) == \phi)$ and $(w_{um_d} < w_{vm_d})$ **then**
    $u.Remove(m)$
  end if
end for

We next detail the evaluation of partial centrality and the detection of overlapped community structure in the following two sections, respectively.

B. Evaluating Partial Centrality

As aforementioned, we mainly focus on the partial centrality of a node relative to the community members. Traditional solutions for computing node centrality are not applicable, due to the unknown number of neighbors and vulnerable end-to-end path in opportunistic networks. To deal with this issue, we use the technology of Principal Component Analysis (PCA) [17] to evaluate the partial centrality metric, and correspondingly, to detect the overlapped community structure.

Principal Component Analysis: Principal component analysis is a powerful tool to extract relevant information from a data set by filtering noise and redundant data. This relevant information reveals the hidden, simplified structures underlying the data set. We generalize the principle of PCA as follows.

Suppose that a node u has built the matrix W from its view of the DAG (please refer to Section II.B), and the matrix W has been centralized (i.e., subtract the corresponding mean from each of columns). Let $C_W = W^T W/(n-1)$ denote the covariance matrix of W. Let us further diagonalize the C_W as

$$P^T C_W P = \Lambda \tag{7}$$

where $\Lambda = diag(1, 2, ..., n)$ and P is a normalized orthogonal matrix. Let x_i be the eigenvectors of C_W and λ_i the corresponding eigenvalues, and $\lambda_1 \geq \lambda_2 \geq ...\lambda_n$. We can see from Fig.2 that the row vector $\alpha_u(\alpha_{u1}, \alpha_{u2}, ..., \alpha_{un})$ denotes the distribution of node u in the n-dimensional spectral space, and the column vector $x_i(\alpha_{1i}, \alpha_{2i}, ..., \alpha_{ni})$ denotes the coordinates of all of the nodes in the ith dimension of the spectral space. In addition, once we get the orthogonal matrix P, we generally select the top k-dimensional spectral space $(x_1, x_2, ..., x_k)$ as the $principal\ component$ of W, since the corresponding top k eigenvalues dominate the spectral graph features [21]. Algorithm 2 describes the above computation process and Table.I lists the main notations used in the paper.

$$\begin{array}{cccccccccc} & x_1 & & x_i & & x_k & x_{k+1} & & x_n \\ P = & \left[\begin{array}{ccccccccc} a_{11} & \cdots & a_{1i} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ a_{u1} & \cdots & a_{ui} & \cdots & a_{uk} & a_{u,k+1} & \cdots & a_{un} \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{ni} & \cdots & a_{nk} & a_{n,k+1} & \cdots & a_{nn} \end{array}\right] \end{array}$$

Fig. 2. The spectral space of W and its vector representation.

TABLE I
THE MAIN NOTATIONS USED IN THE PAPER

NOTATION	Explanation
G	The decayed aggregation graph
W	The adjacent matrix of graph G
w_u	The row vector of matrix W
C_W	The covariance matrix of W
P_{k+1}	The noise components of W
P_k	The principal components of W
P	The eigenvector decomposition of C_W
W_k	The dimensionality reduction matrix of W
C_{W_k}	The covariance matrix of W_k
Λ	The diagonal matrix of C_W
λ_i	The ith eigenvalue of C_W
x_i	The ith eigenvector of C_W
α_u	The distribution of node u in the n-dimensional spectral space
$\alpha_u^{1,k}$	The signal distribution of α_u
$\alpha_u^{k+1,n}$	The noise distribution of α_u

Algorithm 2 PCA

Input: an adjacency matrix W of DAG
Output: orthogonal matrix P and diagonalized matrix Λ
$W = centralized(W)$
$C_W = cov(W)$
$[P, \Lambda] \longleftarrow eigs(C_W, n)$

Mathematically, let matrix $P_k = (x_1, x_2, ..., x_k)$ and $\Lambda_k = diag(1, 2, ..., k)$. Let $\alpha_u^+ = (|\alpha_{u1}|, |\alpha_{u2}|, ..., |\alpha_{ui}|, ..., |\alpha_{uk}|)$, where $|\alpha_{ui}|$ denotes the absolute value of α_{ui}, we have

Lemma 2: For a given decayed aggregation graph G with k communities, the matrix P_k is the projection matrix, the vector α_u^+ presents the likelihood of node u's attachment to such k communities.

Proof: Let W_k denote the dimensionality reduction matrix of W and the matrix C_{W_k} be the covariance matrix of W_k. Based on the theory of PCA, C_{W_k} should been diagonalized as well, we have

$$C_{W_k} = \frac{W_k^T W_k}{n-1} = \Lambda_k \tag{8}$$

On the other hand, from Eq.(7), we get

$$P^T C_W P = \Lambda \Rightarrow P_k^T C_W P_k = \Lambda_k \tag{9}$$

Replace Λ_k with Eq.(8) and C_W with $W^T W/(n-1)$, respectively, we have

$$\frac{W_k^T W_k}{n-1} = P_k^T C_W P_k \Rightarrow \frac{W_k^T W_k}{n-1} = \frac{P_k^T W^T W P_k}{n-1}$$

Multiply both sides by $(n-1)$ and use the substitution of $P_k^T W^T = (W P_k)^T$, we get

$$W P_k = W_k \tag{10}$$

Hence, we conclude that $|\alpha_{ui}|$ is the projection length of node u in community i. ■

Theorem 2 (Node u's partial centrality $\mathcal{I}$): Let PC_u^i denote the partial centrality of node u with respect to a community i, we have

$$PC_u^i = |\alpha_{ui}|\,\lambda_i \tag{11}$$

Proof: From the *Lemma 2*, we know that the likelihood of node u's attachment to the community i equals to $|\alpha_{ui}|$, and from the spectral graph theory [21], it has been shown that the eigenvalue λ_i indicates the strength of community i in the graph G. Hence, we get $PC_u^i = |\alpha_{ui}|\,\lambda_i$. ■

Similarly, if node u belongs to multiple communities, we have the following $Theorem\ 3$.

Theorem 3 (Node u's partial centrality $\mathcal{II}$): Let PC_u denote the partial centrality of node u and k_u denote the number of communities including node u ($k_u \leq k$), we have

$$PC_u = \sum_{i=1}^{k_u} |\alpha_{ui}|\,\lambda_i \tag{12}$$

C. Detecting the Overlapped Community Structure

Cutting a graph into small clusters has been studied widely. We test the k-means, one of the most well-known clustering algorithms [22], by extending it into an overlapped community structure. The advantage of the k-means algorithm compared to other methods such as CNM [23] and k-clique [24] is that it does not need to know the neighbor relationship between nodes, and only requires the adjacent matrix of a weighted graph such as the DAG, while the CNM and k-clique are more appropriate to a binary graph. In addition, based on the technology of PCA discussed above, we can confidently determine the number of communities, the initial elements for each community and the termination condition, three issues strongly affecting the performance of k-means. We next discuss how to detect the overlapped community structure based on the refined k-means.

Determining k, the number of communities: PCA provides a roadmap to reduce a confusing data set to a lower dimension that retains the main features of the original data set. The rationale behind this is that the eigenvalues of a network, play a big role in many important graph features. It has been shown that the maximum degree, clique number, and even the randomness of a graph are all related to λ_1. In general, we select the top k eigenvectors to denote the main structures of the graph, where the value of k satisfies

$$\sum_{i=1}^{k} \lambda_i \Big/ \sum_{j=1}^{n} \lambda_j \geq R \tag{13}$$

and the ratio R usually belongs to the interval [0.7, 0.9] [17]. In this paper, we set $R = 0.85$, we believe it is enough to characterize the main structures of a network (please refer to the Section IV.A).

Identifying the noise nodes: PCA divides a network into two different parts: 1) the principal components P_k, and 2) the opposite P_{k+1}, where the $P_{k+1} = (x_{k+1}, x_{k+2}, ..., x_n)$, as shown in Fig.2. We call the latter noise components of the network. And accordingly, we divide the row vector α_u by $\alpha_u^{1,k}(\alpha_{u1}, \alpha_{u2}, ..., \alpha_{uk})$ and $\alpha_u^{k+1,n}(\alpha_{u,k+1}, \alpha_{u,k+2}, ..., \alpha_{un})$, the signal and noise of the node u. The following $definition\ 2$ helps to identify whether a node is a noise node or not.

Definition 2 (Node u's $signal{-}to{-}noise\ ratio\ SNR_u$): $SNR_u = \sum_{i\in[1,k]} (\lambda_i \alpha_{ui})^2 \Big/ \sum_{j\in[k+1,n]} (\lambda_j \alpha_{uj})^2$.

From the *Theorem 2*, we know that the node u's partial centrality relative to community i is $|\alpha_{ui}|\,\lambda_i$, which is also the amplitude of node u's signal in the ith dimensional spectral space. Hence, the signal energy e^u_{signal} of node u can be presented as $e^u_{signal} = \sum_{i\in[1,k]} (\lambda_i |\alpha_{ui}|)^2 = \sum_{i\in[1,k]} (\lambda_i \alpha_{ui})^2$, and the noise strength e^u_{noise} equals to $\sum_{j\in[k+1,n]} (\lambda_j \alpha_{uj})^2$. Based on $definition$ 2, we propose the following definition.

Definition 3 (Noise Nodes): The node u is a noise node if its SNR_u satisfies $SNR_u < 1$.

Determining the initial elements for each community: After we have ascertained the number of communities and excluded the noise nodes, the next step is to determine the initial centroid $m_i (i = 1, 2, ..., k)$ for each community. We select the node u, s.t. $\max |\alpha_{ui}| \,(u = 1, 2, ..., n)$ for each eigenvector x_i, as the initial node of community i, and set $m_i = \alpha_u$. Algorithm 3 describes this procedure.

Algorithm 3 The Initial Centroid

```
Input: P_k, maxValue ← 0, v ← 0
Output: m_i (i = 1, 2, ..., k)
for i=1 to k do
    maxValue = |α_1i|
    v ← 1 {Tracking who is the maximum}
    for u=2 to n do
        if |α_ui| > maxValue ∧ u is not a noise node then
            maxValue = |α_ui|, v ← u
        end if
    end for
    C_i ← C_i ∪ {v}, m_i ← α_v
    Com(v) ← Com(v) ∪ {i}
end for
```

Termination condition of k-means: Suppose all of the non-noise nodes have been clustered, and the m_i is updated by

$$m_i = \left(\sum\nolimits_{u \subset C_i} \alpha_u\right) / n_i \tag{14}$$

where n_i is the number of nodes only belonging to $\mathcal{C}_i$ (i.e., node u is a strong node, see $Definition$ *4*). k-means is characterized by minimizing the sum of squared errors,

$$J = \sum_{i=1}^{k} \sum_{u \subset C_i} (\alpha_u - m_i)^2 \tag{15}$$

It has been shown that the standard iterative method to k-means suffers seriously from the local minima problem, because of the greedy nature of the update strategy. Fortunately, the $Theorem$ *4* guarantees the PCA-based k-means is immune to this problem.

Theorem 4 (Theorem 3.2 of [25]): Minimizing J is equivalent to maximizing $trace(P^T C_W P)$ (please refer to Eq. (19) of [25]), and $\max trace(P^T C_W P) = \lambda_1 + \lambda_2 + ... + \lambda_k$.

In other words, the PCA-based k-means has reached the optimal performance once we cluster all of the non-noise nodes for the first time.

Detecting the overlapped community structure: In this paper, we allow a non-noise node to join multiple communities, and classify them into the following two categories 1) strong nodes and 2) bridging nodes.

Definition 4 (Strong Nodes): A node u is a strong node if it only belongs to one community.

Definition 5 (Bridging Nodes): A node u is a bridging node if it joins two or multiple communities.

We now discuss how to online identify the strong nodes and bridging nodes based on the following steps.

(1) Clustering nodes: For any node u, we compute the distance between itself and the centroid m_i, $dist(\alpha_u, m_i)$, and select i, s.t. $\min dist(\alpha_u, m_i)$ $(i = 1, 2, ...k)$ as the community node u belongs to, where,

$$dist(\alpha_u, m_i) = \theta(u, i) = \arccos \frac{\alpha_u {m_i}^T}{\|\alpha_u\|_2 \|m_i\|_2}$$

and $\theta(u, i)$ denotes the angle between α_u and m_i. We label node u a strong node and update m_i by Eq. (14). For other $\theta(u, j)$ $(j \neq i, j = 1, 2, ...k)$, we label node u a bridging node belonging to community j, if and only if the $\theta(u, j) \in [\pi/4 - \varphi, \pi/4 + \varphi]$ (please refer to $Theorem$ *5*), where φ is the overlapped coefficient. Algorithm 4 describes the clustering procedure.

(2) Adjusting the categories of nodes: After step (1) finishes, the explicit community structure has been detected together with the blurred labels of nodes. Because some of nodes with "strong" labels may share multiple communities, those with "bridging" only belong to one community, or even such are labelled with strong and bridging simultaneously. To this end, we need to re-classify each node. Let $|Com(u)|$ denote the number of communities node u belongs to. Algorithm 5 presents the adjusting process.

Determining the overlapped interval $[\pi/4 - \varphi, \pi/4 + \varphi]$: This section focuses on why we set the overlapped interval $[\pi/4 - \varphi, \pi/4 + \varphi]$.

Algorithm 4 Clustering nodes

for u=1 to n **do**
  for i=1 to k **do**
    Computing $dist(\alpha_u, m_i)$
  end for
  Selecting i, s.t. $\min \theta(u, i)$ $(i = 1, 2, ...k)$
  $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{u\}$
  $Com(u) \leftarrow Com(u) \cup \{i\}$
  Updating m_i
  // Identifying bridging nodes
  for Other $\theta(u, j)$ $(j \neq i, j = 1, 2, ...k)$ **do**
    if $\theta(u, j) \in [\pi/4 - \varphi, \pi/4 + \varphi]$ **then**
      $\mathcal{C}_j \leftarrow \mathcal{C}_j \cup \{u\}$
      $Com(u) \leftarrow Com(u) \cup \{j\}$
    end if
  end for
end for

Algorithm 5 Adjusting node's categories

for u=1 to n **do**
  if $|Com(u)| \geq 2 \wedge u$ is a strong node **then**
    u is a bridging node
  end if
  if $|Com(u)| == 1 \wedge u$ is a bridging node **then**
    u is a strong node
  end if
end for

Lemma 3: Strong nodes from k communities form k quasi-orthogonal lines in the spectral space.

Proof: From *Definition 4* and the clustering process mentioned above, we know that the centroid m_i $(m_{1i}, m_{2i}, ..., m_{ni})$ can approximately present the line formed by strong nodes within the ith community (please refer to Eq. (14)). On the other hand, the virtual centroid vector m_i should be close to eigenvector x_i. This is mainly because $m_i \approx \widetilde{m_i} = \left(\sum\nolimits_{u \subset C_i} \alpha_{ui}\right) / n_i$, as α_{ui} is the dominant part of α_u. Hence, $\widetilde{m_i}$ locates in the line formed by the eigenvector x_i. We get the conclusion as different eigenvectors are linearly independent. ■

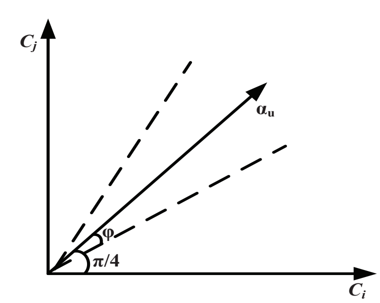


Fig. 3. Overlapped coefficient and confidence interval.

Theorem 5 (Identifying Bridging Nodes): A node u can join two communities i and j if $\theta(u, i), \theta(u, j) \in [\pi/4 - \varphi, \pi/4 + \varphi]$.

Proof: From Lemma 3, bridging nodes are not significantly close to any of the lines. We believe that a "theoretic" bridging node should be located along the diagonal of a 2-dimensional space as shown in Fig.3. That is, the angle between α_u and C_i should equal to $\pi/4$. We set the confidence interval $[\pi/4-\varphi, \pi/4+\varphi]$ in this paper when considering the practical situations, and develop an overlapped coefficient φ to adaptively adjust the radian of interval. ∎

IV. Data-sets and Experimental Results

We first analyze the overlapped community structures underlying the data-sets we used, and then compare the performance of OFPC with two state-of-the-art works: Bubble and Prophet [4] together with the Epidemic [31] and Direct Contact [3] algorithms as benchmarks. Bubble is a well-known social-based forwarding algorithm and Prophet is currently an IETF draft [30]. Results of Epidemic and Direct algorithms provide us the upper and lower bounds of important performance evaluation metrics: mean delivery delay, cost and packet delivery ratio.

A. Data-sets

We use the following three real data-sets gathered by [26] [27], referred to as North Carolina State Fair, NCSU, and KAIST. The characteristics of these data-sets such as intra/inter-contact distribution have been explored in several studies (e.g., [26] [27]) and applied into different scenarios [28] [29]. Interestingly, by analyzing these traces, we find that they cover a rich diversity of environments ranging from well connected scenario (Statefair) to quite sparse situation (NCSU). The general statistics of the three data-sets are summarized in Table II.

TABLE II
STATISTICS OF COLLECTED REAL TRACES FROM THREE SITES

Site	No. of trajectories	volunteers	start date / end date
Statefair	19	18	2006-10-24 / 2007-10-21
NCSU	35	20	2006-08-26 / 2006-11-16
KAIST	92	34	2006-09-26 / 2007-10-03

TABLE III
STATISTICS OF COMMUNITY STRUCTURE UNDERLYING THE THREE SCENARIOS ($\beta = 1, \varphi = 0.027 \approx 5^o$)

Site	maximum	minimum	mean	variance
Statefair	7	3	5.14	0.8661
NCSU	11	4	7.97	3.5931
KAIST	9	1	5.25	6.1905

Overlapped community structures underlying the data-sets: Fig.4 illustrates the number of communities hidden behind the three scenarios at different moments. We observe that Statefair exhibits a more stable topology, compared to KAIST and NCSU. The variance of the number of communities is 0.8661 at Statefair, while those at KAIST and NCSU are 6.1905 and 3.5931, respectively, as shown in Tab.III, where the maximum, minimum and mean of the number of communities are presented as well. In addition, Tab.IV presents the average ratio of the noise, bridging and strong nodes of the three scenarios. We observe that NCSU shows a poorest connectivity and there exists 7 ($\lfloor 21.23\% \times 35 \rfloor$) noise nodes. At the same time, we notice that there indeed exist overlapped community structures underlying these data-sets, since there exist 6 ($\lfloor 7.45\% \times 92 \rfloor$) bridging nodes at KAIST, 3 ($\lfloor 11.02\% \times 35 \rfloor$) bridging nodes at NCSU, and 2 ($\lfloor 11.39\% \times 19 \rfloor$) bridging nodes at Statefair, respectively.

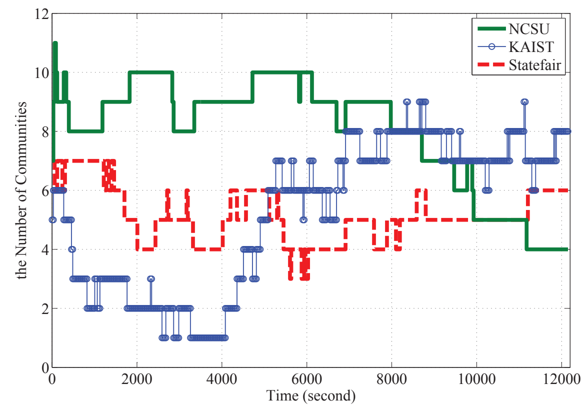


Fig. 4. The number of communities at different moments

TABLE IV
AVERAGE RATIO OF THE NUMBER OF NOISE, BRIDGING AND STRONG NODES ($\beta = 1, \varphi = 0.027 \approx 5^o$)

Site	noise nodes (%)	bridging nodes (%)	strong nodes (%)
Statefair	5.36	11.39	83.25
NCSU	21.23	11.02	67.75
KAIST	9.69	7.45	82.86

B. Simulation Setup

We utilize the aforementioned three real data-sets to test the premise of forwarding scheme based on social structures. For each data-set, one randomly chosen source node sends a message to one randomly chosen destination node, and total 1000 messages are generated. The nodal transmission range is set to $250m$, a typical value of WiFi, and the emulation results are the average over 50 runs for statistical confidence. In addition, we compare OFPC against the optimized versions of Bubble and Prophet. We use an offline method to compute the betweenness centrality of each node for Bubble (i.e., we first flood a large number of messages in the network, and count the number of times a node acts as a relay for other nodes on all the shortest paths [13]), and take the default experimental parameters for Prophet [4].

C. Performance Evaluation

Mean delivery delay (MDD): Fig.5 illustrates the performance of mean delivery delay within different message TTLs. It's obvious to see that OFPC expedites the dissemination speed of message. For example, it achieves up to a 70% improvement in MDD over Prophet and 40% over Bubble at Statefair (Fig.5.(a)). The reason behind this is that OFPC exploits the partial centrality metric to make forwarding decisions, this novel metric provides us a fine-grained level

of characterizing the relations between nodes, thus, it helps to choose the more qualified relays than the centrality-based scheme does.

Cost: Fig.6 clarifies that OFPC has the best performance in term of cost as well. For example, at Statefair, OFPC helps considerably in reducing up to $2\times$ and $3\times$ overhead in Bubble and Prophet. Even at NCSU (Fig.6.(b)), the very sparse scenario, OFPC still outperforms Bubble and Prophet. This is mainly because 1) the partial centrality metric helps to improve the delivery delay, and in turn to reduce the cost, and 2) we exclude the noise nodes from the relay candidates, as they are isolated and far away from the community members (please refer to Section III.A).

Packet delivery ratio (PDR): Fig.7 presents the performance of packet delivery ratio. We can see that, in general, OFPC achieves a similar delivery ratio to Bubble, and both of them outperform the Prophet.

V. RELATED WORK

In the past, a lot of opportunistic forwarding algorithms have been proposed. we classify them into the following two categories based on the contexts they used.

Physical contact based: A. Vahdat and D. Becker first proposed an epidemic forwarding style for partially connected ad hoc networks [31]. They tried to grasp each forwarding opportunity thus guaranteeing a high packet delivery ratio while consuming more system resources as well. This deficiency has motivated researchers to develop other forwarding mechanisms (e.g, [4] [7] [8]). For most of them, the networking performance depends heavily on the contexts they utilized to identify "the best" relay node to destination. For example, A. Lindgren et al. [4] presented Prophet, a probabilistic routing protocol for opportunistic networks. They exploited past contact moments to predict the probability of future encounters. Similarly, J. Leguay et al. [7] proposed MobySpace, a high-dimensional Euclidean space constructed by the past contact locations. Apparently, this scheme reduces the overhead but increases the delivery delay.

Social contact based: Noting that the aforementioned physical contact based scheme does not consider the social structures evolving from human activities. Whereas, with the recent popularization of personal hand-held mobile devices, human mobility gradually plays a critical role in networking performance, as human walks show a strong spatiotemporal correlation (e.g., clustering) [9] [10], instead of purely random motions. Considering this fact, researchers have recently focused on the influence of social structure on opportunistic communication. For instance, E. Daly, P. Hui and A. Mtibaa et al. [12] [13] [14] further exploited social structures such as centrality/similarity metric to make forwarding decisions. Messages will be forwarded to such nodes with relatively high centrality/similarity metrics to increase the probability of finding better relays to the final destination. For example, in SimBet [12], each node evaluates its centrality and similarity metrics based on the ego network technology, and a message it carries is either forwarded to nodes having higher similarities with the destination node, or stays with the most central node. Similarly, in Bubble [13], a message is relayed across nodes with increasing centrality metrics, until it enters into the range of the destination community. In addition, A. Mtibaa et al. proposed PeopleRank [14], which exploited PageRank [16] algorithm to evaluate node centrality, and a message is only forwarded to such nodes with higher centralities than the current carriers.

Obviously, the main difference between our work and the state-of-the-art works is that we explore the impact of partial centrality metric on performance of opportunistic routing. We believe that this novel metric helps to make informed forwarding decisions.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose OFPC, a partial centrality metric based forwarding algorithm, to improve the performance of opportunistic routing. We first formulate the strength of relationship between nodes as a Decayed Sum Problem, and use a Decayed Aggregation Graph to model the opportunistic network. We then present an online method to evaluate node partial centrality by exploiting the theory of principal component analysis. Third, we detect the overlapped community structure combining the technology from PCA and k-means algorithm. We finally validate the effectiveness of our method by trace-driven simulation. One significant topic for future work is to extend the partial centrality metric to other important applications such as recommendation system and worm containment in online social networks.

REFERENCES

[1] Kevin Fall, Stephen Farrell. DTN: An Architectural Retrospective. IEEE Journal on Selected Areas in Communications, 2008, 26(5):828-836.

[2] Marco Conti, Mohan Kumar. Opportunities in Opportunistic Computing. Computer, 2010, 43(1):42-50.

[3] R. C. Shah, S. Roy, Sushant Jain, et al. Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Networks, 2003,1(2$\sim$3):215-233.

[4] A. Lindgren, A. Doria, O. Schelen. Probabilistic Routing in Intermittently Connected Networks. Lecture Notes in Computer Science, 2004,3126:239-254.

[5] A. Balasubramanian, B. N. Levine, A. Venkataramani. Replication Routing in DTNs: A Resource Allocation Approach. IEEE/ACM Trans. on Networking, 2010, 18(2):596-609.

[6] G. Dinan, K. Thomas, P. Alexandre, et al. Scoop: Decentralized and Opportunistic Multicasting of Information Streams. In Proc. MobiCom 2011, ACM, pp. 169-180.

[7] J. Leguay, T. Friedman, V. Conan. Evaluating Mobility Pattern Space Routing for DTNs. In Proc. Infocom 2006, IEEE, pp.1-10.

[8] V. Erramilli, A. Chaintreau, M. Crovella, et al. Diversity of forwarding paths in pocket switched networks. In Proc. IMC 2007, ACM, pp. 161-174.

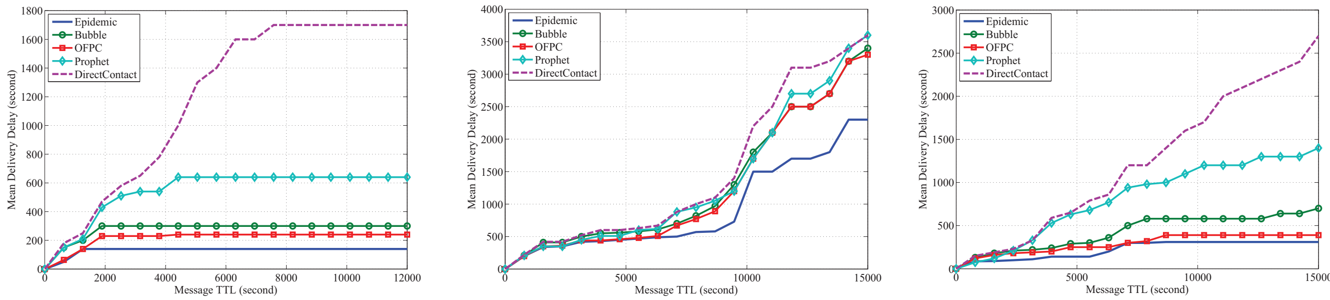
[9] C. Augustin, P. Hui, C. Jon, et al. Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms. IEEE Transactions on Mobile Computing, 2007, 6(6):606-620.

[10] Wei Gao, Guohong Cao. Fine-Grained Mobility Characterization: Steady and Transient State Behaviors. In Proc. MobiHoc 2010, ACM, pp.61-70.

[11] P. V. Marsden. Egocentric and sociocentric measures of network centrality. Social Networks, 2002, 24:407-422.

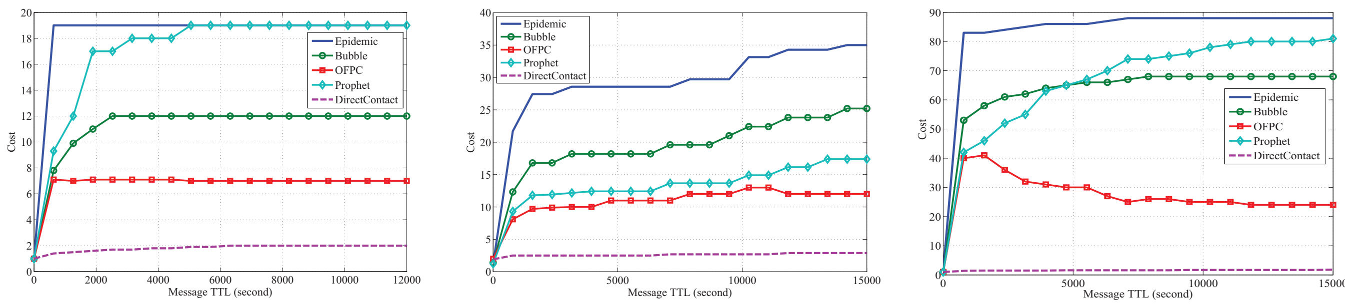
[12] E. Daly, M. Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In Proc. MobiHoc 2007, ACM, pp.32-40.

[13] Pan Hui, J. Crowcroft, E. Yoneki. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In Proc. MobiHoc 2008, ACM, pp.241-250.



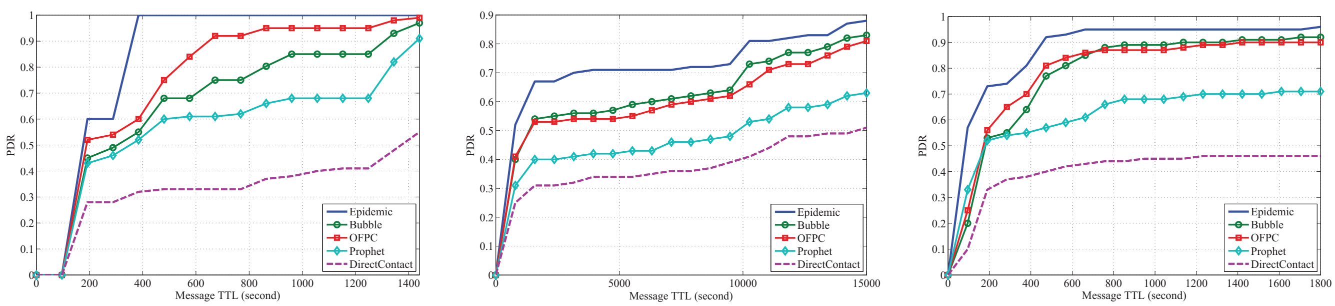
(a) Statefair (b) NCSU (c) KAIST

Fig. 5. Mean delivery delay within different message TTLs



(a) Statefair (b) NCSU (c) KAIST

Fig. 6. Cost: the average number of nodes infected by a message within different message TTLs



(a) Statefair (b) NCSU (c) KAIST

Fig. 7. Packet delivery ratio within different message TTLs

[14] A. Mtibaa, M. May and C. Diot et al. PeopleRank: Social Opportunistic Forwarding. In Proc. Infocom 2010, IEEE, pp.111-115.

[15] A. K. Pietilainen, C. Diot. Dissemination in Opportunistic Social Networks: the Role of Temporal Communities. In Proc. MobiHoc 2012, ACM, pp. 165-174.

[16] S. Brin, L. Page. The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst., 1998, 30(1-7):107-117.

[17] I. Jolliffe. Principal component analysis. Springer, 2nd ed., 2002.

[18] E. Cohen, M. J. Strauss. Maintaining Time-Decaying Stream Aggregates. Journal of Algorithms, 2006, 59(1):19-36.

[19] FREEMAN, L. C. Centrality in social networks conceptual clarification. Social networks, 1979, 215-239.

[20] T. Karagiannis, J-Y. Le. Boudec, Milan Vojnovic. Power Law and Exponential Decay of Inter Contact Times between Mobile Devices. In Proc. MobiCom 2007, ACM, pp.183-194.

[21] Fan R. K. Chung. Spectral Graph Theory. American Mathematical Society, 1997.

[22] S. Fortunato. Community detection in graphs. Physics Reports, 2010, 486:75-174.

[23] A. Clauset, M. Newman, C. Moore. Finding community structure in very large networks. Physical Review E., 2004, 70:066111.

[24] G. Palla, I. Derenyi, I. Farkas, et al. Uncovering the overlapping community structure of complex networks in nature and society. Nature, 2005, 435:814-818.

[25] C. Ding, X. He. K-means Clustering via Principal Component Analysis. In Proc. ICML 2004, ACM, pp. 225-232.

[26] K. Lee, S. Hong, S. J. Kim. Demystifying Levy Walk Patterns in Human Walks. Technical Report, CSC, NCSU, 2008.

[27] I. Rhee, M. Shin, S. Hong et al. On the Levy-walk Nature of Human Mobility. In Proc. Infocom 2008, IEEE, pp. 924-932.

[28] S. Kaveevivitchai,H. Esaki. Independent DTNs Message Deletion Mechanism for Multi-copy Routing Scheme. In Proc. AINTEC 2010, ACM, pp. 48-55.

[29] S. Rallapalli, Lili Qiu, Yin Zhang et al. Exploiting Temporal Stability and Low-Rank Structure for Localization in Mobile Networks. In Proc. MobiCom 2010, ACM, pp. 161-172.

[30] DTN Research Group. Extensions of Probabilistic Routing Protocol for Intermittently Connected Networks. Downloaded from http://www.ietf.org/id/draft-softgear-dtnrg-eprophet-01.txt, May, 2012.

[31] Vahdat A, Becker D. Epidemic routing for partially connected ad hoc networks, cs2200006. Durham, North Carolina : Duke University,2000.